

National Aeronautics and
Space Administration

SLS-PLAN-156
VERSION 3
RELEASE DATE: OCTOBER 11, 2016

SPACE LAUNCH SYSTEM PROGRAM (SLSP)
FLIGHT SOFTWARE APPLICATION
SOFTWARE ASSURANCE PLAN (SAP)

Approved for Public Release; Distribution is Unlimited.
The electronic version is the official approved document.
Verify this is the correct version before use.

HISTORY PAGE

Status	Version No.	Change No.	Description	Release Date
Release	1		Added document number; Initial Release of SLS DRD 1406SM-014. 4511 Concurrence on file.	5/8/12
	2		Release of SLS DRD 1406SM-014. MSFC 4511 Concurrence on file. Incorporate SA SCR #227, SA SCR #228, Incorporate Stages PDR comments, Incorporate SLSP PDR comments, Remove TBD, Editorial corrections; Remove Appendices; Manpower Estimates Updated. Resolved CM Document Audit findings. 4511 Concurrence on file.	12/9/13
	3		Release of SLS DRD 1406SM-014. MSFC 4511 Concurrence on file. ES50 System Review Board Approved 10/11/2016 Incorporate CR#UPD1350 and Task Team comments.	10/11/16

NOTE: Updates to this document, as released by numbered changes (Change XXX), are identified by a black bar on the right margin.

TABLE OF CONTENTS

PARAGRAPH **PAGE**

1.0 INTRODUCTION 8
1.1 Purpose 8
1.2 Scope 8
1.3 Change Authority/Responsibility 9
1.4 Computer Software Configuration Items – Identification (CSCI) 9
1.5 Space Launch System (SLS) Overview 10
1.6 Software Development Facility (SDF) Overview 10
1.7 Document Overview 10
1.8 Relationships to Other Plans 12
2.0 DOCUMENTS 12
2.1 Applicable Documents 12
2.2 Reference Documents 13
3.0 MANAGEMENT 14
3.1 Software Assurance Organization 14
3.1.1 SLS FSW Software Assurance Structure 14
3.2 Resources 15
3.2.1 Facilities and Equipment 15
3.2.2 Personnel 16
3.3 Software Assurance Tasks 16
3.3.1 Task: Perform Software Classification Assessment 17
3.3.2 Task: Perform Software Assurance Classification Assessment 17
3.3.3 Task: Perform Independent Verification and Validation (IV&V) Assessment 17
3.3.4 Task: Perform Software Assurance Documentation Development and Maintenance 17
3.3.5 Task: Evaluate Deviations and Waivers 18
3.3.6 Task: Review and Approve Deliverable Software Products 18
3.3.7 Task: SA Problem Reporting and Corrective Action 18

3.3.8 Task: Evaluate Project Planning Process 18
3.3.9 Task: Evaluate System Requirements Analysis Process 19
3.3.10 Task: Evaluate Software Requirements Process 19
3.3.11 Task: Evaluate Software Design Process 20
3.3.12 Task: Evaluate Coding and Unit Testing Process 20
3.3.13 Task: Evaluate Unit Integration and Testing, CSCI Qualification Testing, Integration and Testing, and System Qualification Testing....21
3.3.14 Task: Non-Deliverable Software 22
3.3.15 Task: Evaluate Non-Developmental Software 22
3.3.16 Task: Verify Software Configuration Audits 22
3.3.17 Task: Verify Implementation of Requirements Management 23
3.3.18 Task: Verify Implementation of Software Project Planning 23
3.3.19 Task: Verify Implementation of Software Project Monitoring and Control 23
3.3.20 Task: Verify Implementation of Software Supplier Agreement Management 24
3.3.21 Task: Verify Implementation of Measurement and Analysis 24
3.3.22 Task: Verify Implementation of Software Configuration Management 24
3.3.23 Task: Verify Implementation of Requirements Development 24
3.3.24 Task: Verify Implementation of Technical Solution 25
3.3.25 Task: Verify Implementation of Product Integration 25
3.3.26 Task: Verify Implementation of Verification 25
3.3.27 Task: Verify Implementation of Validation 25
3.3.28 Task: Verify Implementation of Integrated Project Management 26
3.3.29 Task: Verify Implementation of Risk Management 26
3.3.30 Task: Verify Implementation of Decision Analysis and Resolution 26
3.4 Stakeholder Responsibilities 27
3.5 Schedule 30
4.0 DOCUMENTATION 30
5.0 STANDARD, PRACTICES, CONVENTIONS, AND METRICS 31
5.1 Participants in Software Assurance 31
5.2 Metrics 31

6.0 REVIEWS 31
6.1 Technical Reviews 31
6.2 Management Reviews 31
6.3 Peer Reviews 32
7.0 TEST 32
8.0 SA PROBLEM REPORTING AND RESOLUTION 32
8.1 Product Audit Reporting 32
8.1.1 Product Audit Corrective Action Implementation 33
8.1.2 Product Audit Unresolved Concerns/Issues Escalation 33
8.2 Process Audit Reporting 34
8.2.1 Process Audit Findings/Observations Corrective Action Implementation 34
8.2.2 Process Audit Unresolved Findings Escalation 34
8.3 Peer Review Reporting 35
8.3.1 Peer Review Defects/Typos Corrective Action Implementation 35
8.3.2 Peer Review Unresolved Defect Escalation 35
8.4 Milestone Review Reporting 36
8.4.1 Milestone Review RID Corrective Action Implementation 36
8.4.2 Milestone Review RID Escalation 36
9.0 TOOLS, TECHNIQUES, AND METHODOLOGIES 37
9.1 Tools 37
9.2 Techniques 38
9.3 Methodologies 38
10.0 MEDIA CONTROL 39
11.0 SUPPLIER CONTROL 39
12.0 RECORDS, COLLECTION, MAINTENANCE, AND RETENTION 39
12.1 Records 39
12.1.1 Software Assurance Plan 39
12.1.2 Process Audit Reports 39
12.1.3 Assessments 40

12.1.4 Status Reports 40
12.1.5 Software Reliability Analysis 40
12.2 Collection 41
12.3 Distribution 41
12.4 Maintenance 41
12.5 Retention 42
13.0 TRAINING 42
14.0 RISK MANAGEMENT 42
15.0 SOFTWARE SAFETY 43
16.0 SOFTWARE RELIABILITY 43
16.1 Introduction 43
16.1.1 Purpose and Goal 43
16.1.2 Software Reliability Methodology 43
16.2 Software Life Cycle Reliability Tasks 45
16.2.1 Software Reliability in the Requirements Phase 45
16.2.2 Software Reliability in the Preliminary Design Phase 46
16.2.3 Software Reliability in the Detailed Design Phase 46
16.2.4 Software Reliability in the Software Implementation Phase 47
16.2.5 Software Reliability in the Software Test Phase 48
16.2.6 Software Reliability in the Operations and Maintenance Phase 49
17.0 INDEPENDENT VERIFICATION AND VALIDATION 50

APPENDIX

APPENDIX A ACRONYMS AND ABBREVIATIONS AND GLOSSARY OF TERMS 51
APPENDIX B COMPLIANCE MATRIX 53
APPENDIX C OPEN WORK 57

TABLES

TABLE 1-1. SOFTWARE LIFE CYCLE ACTIVITIES 9

TABLE 3-1. SOFTWARE ASSURANCE MANPOWER ESTIMATES....16

TABLE 3-2. STAKEHOLDER RESPONSIBILITY MATRIX 27

TABLE 4-1. DELIVERABLE SOFTWARE PRODUCTS 30

TABLE 4-2. NON-DELIVERABLE PRODUCTS....31

TABLE 9-1. SA TOOLS 37

TABLE 12-1. SOFTWARE ASSURANCE DEVELOPED RECORDS....41

TABLE 12-2. DISTRIBUTION OF SA RECORDS....41

TABLE 13-1. SLS PROGRAM SPECIFIC SA TRAINING 42

TABLE 16-1. SOFTWARE RELIABILITY ACTIVITIES BY LIFE CYCLE....45

TABLE C1-1. TO BE DETERMINED ITEMS 57

TABLE C2-1. TO BE RESOLVED ISSUES....57

FIGURES

FIGURE 8-1. ESCALATION OF SA PRODUCT AUDIT UNRESOLVED CONCERN/ISSUE....33

FIGURE 8-2. ESCALATION OF SA PROCESS AUDIT UNRESOLVED FINDING/OBSERVATION 35

FIGURE 8-3. ESCALATION OF SA PEER REVIEW UNRESOLVED DEFECT 36

1.0 INTRODUCTION

1.1 Purpose

The purpose of this plan is to provide a framework for defining the Marshall Space Flight Center (MSFC), Safety and Mission Assurance Directorates (SMA) Software Assurance (SA) organization, tasks and responsibilities. It also provides reference documents and guidelines necessary to perform those SA tasks, and identifies the tools, techniques, and methodologies needed to support these tasks. This plan is in compliance with MSFC DRD STD/QE-SAP, and addresses the programmatic requirements for software assurance found in SLS-RQMT-014, Space Launch System Program (SLSP) Safety and Mission Assurance (SMA) Requirement Document.

1.2 Scope

This plan establishes the SA tasks to be performed in support of the following MSFC software development efforts for the Space Launch System:

- Flight Computer Application Software (FCAS)
- Green Run Application Software (GRAS)
- I-Load Program (ILOADPGM)
- Exploration Computer Application Software (ECAS)

The above Computer Software Configuration Items (CSCIs) has been classified as a "CLASS A" software effort, per the requirements of NPR 7150.2, NASA Software Engineering Requirements; as such this Software Assurance Plan (SAP) represents an overall Full/High SA effort. Additionally, these CSCIs has been determined to be safety critical per the Software Safety Litmus Test defined in NASA-STD-8739.8, Software Assurance Standard.

Additionally, this plan covers the following tools, which are used to support the development, test, and operation of the above CSCIs.

- MAFGEN Class C, safety critical
- PCFExtract Class C, safety critical
- PCFEdit Class C, safety critical
- PCFConstraint Class C, safety critical
- PCFMerge Class C, safety critical
- ExtractPARs Class C, safety critical
- PCFSplitter Class C, safety critical

- Test Diagnostic & Analyses Tool (TDAT) Class C, safety critical
- PARCompare Class C, non-safety critical
- FSWScan Class D, non-safety critical
- SRSUtil Class D, non-safety critical

The Safety and Mission Assurance (SMA) Organizational Instructions (OIs) defined herein will be used to examine all associated software products and processes to determine compliance with technical, performance and programmatic requirements defined by the software documentation products identified in the SLS-PLAN-073, Space Launch System Program Flight Software Development Plan, and cited in Section 4, Table 4-1 of this plan for convenience.

Table 1-1 shows the software life-cycle activities to which this plan applies.

Table 1-1. Software Life Cycle Activities

Life Cycle Activity
Planning
Requirements
Design
Implementation
Integration
Test
Release
Operation and Maintenance
Retirement

1.3 Change Authority/Responsibility

The NASA Office of Primary Responsibility (OPR) for this document is QD34.

Changes to this document shall be controlled at the OPR level using processes defined by the OPR. Compliance with NASA-STD-8739.8, Software Assurance Standard, is addressed in Appendix B.

1.4 Computer Software Configuration Items – Identification (CSCI)

The CSCIs are identified in SLS-PLAN-073, Space Launch System Program Flight Software Development Plan (SDP).

1.5 Space Launch System (SLS) Overview

The SLS will continue the tradition of United States human space flight, serving as follow-on to the Space Shuttle, which was retired in 2011. The SLS is being developed to provide heavy-lift capability to enable human exploration missions beyond low-Earth orbit (BEO), which supports NASA's strategic goal of extending and sustaining human activities across the solar system. The SLS will deliver applicable exploration elements safely to desired orbits, providing sufficient lift mass and volume to execute the desired missions. The SLS will provide:

- An initial lift capability of 70 (Block 1) to 105 (Block 1A) metric tons evolvable to the ultimate capability of 130 metric tons (Block 2)
- A design that is derived from legacy hardware
- Capability to lift the Multi-Purpose Crew Vehicle (MPCV)
- Capability to be a backup system for International Space Station (ISS) cargo and crew delivery

The Space Launch System (SLS) includes the following major elements:

- Stages (Core Stage and Upper Stage)
- Liquid Engines (Core Stage and Upper Stage)
- Boosters (solid or liquid)
- Spacecraft and Payload Integration elements (payload fairings, payload adapters, Interim Cryogenic Propulsion Stage (iCPS)).

1.6 Software Development Facility (SDF) Overview

The Software Development Facility (SDF) provides the facilities for software development, testing, and a simulation environment to perform unit test, initial integration, verification, and validation of the software. Software verification of requirements is conducted in the SDF to the maximum extent for the available hardware. Additionally, the SDF contains the facilities and equipment to perform configuration management and delivery of the flight software.

1.7 Document Overview

This document identifies the SA organization structure, and procedures to be used to perform tasks related to the MSFC QD34 software assurance program as required by MSFC DRD STD/QE-SAP, and NASA-STD-8739.8, while addressing those requirements required by SLS-RQMT-014, Space Launch System (SLS) Program Safety and Mission Assurance (SMA) Requirement Document.

Section 1 identifies the system to which this SAP applies; provides an overview of the system; summarizes the purpose and contents of the SAP; and describes the relationship of this plan to other management plans.

Section 2 lists all documents cited within this plan.

Section 3 describes each major element of the organization that influences the quality of the software and the Software Quality, Software Verification and Validation discipline tasks to be implemented.

Section 4 lists the baseline documents produced and maintained by the developing MSFC organization.

Section 5 identifies the standards, practices, and conventions to be applied on the software development effort.

Section 6 describes the SA reviews to be performed on the software development effort.

Section 7 describes SA participation in testing.

Section 8 describes the problem reporting and corrective action system used during the software life-cycle.

Section 9 describes the SA tools, techniques, and methodologies to be used during the software life-cycle.

Section 10 describes SA's evaluation of media control and software configuration management tool to be used during the software life-cycle.

Section 11 describes the control of supplier software.

Section 12 describes the SA procedures for record collection, maintenance, and retention.

Section 13 describes SA training requirements.

Section 14 describes SA review of the Risk Management process used during the software life-cycle.

Section 15 describes the Software Safety implementation as related to the software development effort.

Section 16 describes the Software Reliability implementation as related to the software development effort.

Section 17 describes the Independent Verification and Validation (IV&V) implementation as related to the software development effort.

Appendix A provides the Abbreviations and Acronyms.

Appendix B provides the Compliance Matrix to NASA-STD-8739.8

Appendix C provides the listing of Open Work

1.8 Relationships to Other Plans

SA evaluation of the software development processes used during all phases, from development through retirement is based on the processes defined in the SLS-PLAN-073, Space Launch System Program Flight Software Development Plan (SDP). The implementation of the SDP, and the applicable SMA Organizational Issuances (OIs), establish the SA evaluation criteria.

2.0 DOCUMENTS

2.1 Applicable Documents

The following documents include specifications, models, standards, guidelines, handbooks, and other special publications. The documents listed in this paragraph are applicable to the extent specified herein.

SLS-PLAN-180	Space Launch System (SLS) Program Risk and Opportunity Management Plan
SLS-RQMT-014	Space Launch System (SLS) Safety and Mission Assurance (SMA) Requirement Document
NASA-STD-8739.8	Software Assurance Standard
NPR 7150.2B	Software Engineering Requirements
SLS-PLAN-073	Space Launch System Program Flight Software Development Plan
SLS-PLAN-074	Space Launch System Program Flight Software Configuration Management Plan (SCMP)
SLS-PLAN-008	Space Launch System (SLS) Program Configuration Management Plan
SLS-PLAN-013	Space Launch System Safety and Mission Assurance Plan
QD-QE-009	Software Assurance Review/Approval of Technical Documents
QD-QE-010	Software Assurance Milestone Review Support
QD-QE-011	Software Quality Assurance Software Audits
QD-QE-012	Software Quality Assurance Support of Formal Software Testing
QD-QE-016	Software Assurance Classification Assessment

ES50-SOP09	Peer Review

2.2 Reference Documents

The following documents contain supplemental information to guide the user in the application of this document.

ES50-SOP01	Process Management
ES50-SOP02	Organizational Training
ES50-SOP03	Planning
ES50-SOP04	Project Management
ES50-SOP05	Requirements
ES50-SOP06	Design, Development, Integration
ES50-SOP07	Test
ES50-SOP08	Audit
ES50-SOP10	Change Management
ES50-SOP11	Trade Study
IEEE 730-2002	IEEE Standard for Software Quality Assurance Plans
NASA-STD-8719.13	Software Safety Standard

3.0 MANAGEMENT

Software Assurance (SA) depends to a great extent upon management support of the Software Assurance Program. The extent to which Software Assurance Program objectives are achieved is directly proportional to the emphasis placed on software assurance in the performance of various management functions. The key factors influencing achievement of software assurance objectives are:

1. Development and implementation of appropriate SA requirements for planning, design, testing, and operation.
2. Sufficient planning, budgeting, and staffing of the SA tasks.
3. Clear assignment of responsibilities for the accomplishment of SA tasks.
4. SA personnel are adequately supervised, trained, and motivated.
5. Internal compliance with SA requirements.
6. Adequate monitoring of the Software Assurance Program, with sufficient feedback to assure satisfaction of requirements.

The focal point for software assurance management resides in the SMA Vehicle Systems Department/QD30, Avionics and Software Branch (QD34), assuring that the Software Assurance Program is properly implemented and supported by all organizational elements.

3.1 Software Assurance Organization

The software assurance organization, an entity independent from the organization creating the software, must ensure and assure that software assurance tasks are performed correctly and to the necessary level, and that records of those tasks are created, analyzed, and maintained. For MSFC this is the Safety and Mission Assurance (SMA) Directorate. One or more software assurance engineers from the Avionics and Software Branch (QD34) may be assigned to work on the SLS Program throughout its life cycle. While these software assurance engineers are a part of the Program and participate in day-to-day activities, perform all of the assurance functions, and attend Program meetings and reviews, they maintain a separate reporting chain through their SMA directorate.

3.1.1 SLS FSW Software Assurance Structure

Figure 3-1 depicts the software assurance organizational structure and its relationship to the SLS Program and the SLS Flight software development organization.

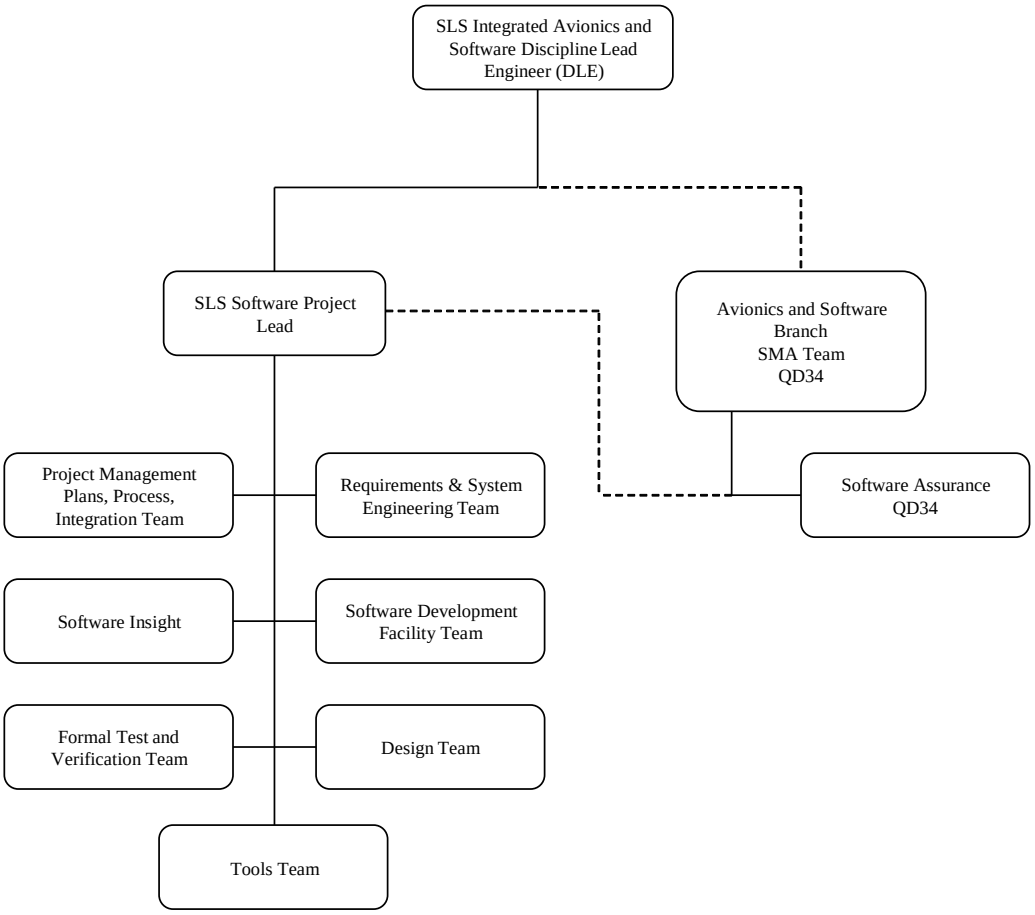


Figure 3-1. SLS FSW Software Assurance Organization Structure

3.2 Resources

3.2.1 Facilities and Equipment

SA will have access to the facilities and equipment as described in the SLS Flight Software Development Plan. SA will have access to computer resources to perform SA tasks, such as process audits, peer reviews, problem reporting, product audits, metric collection, and software defect tracking.

3.2.2 Personnel

Safety and Mission Assurance Directorate will provide trained software assurance personnel for each required software assurance discipline to accomplish the required software assurance tasks over the complete lifecycle of the SLS Program. The SMA department responsible for assuring the assignment of trained software assurance personnel to support the SLS Flight Software development is the Vehicle Systems Department/QD30. The Avionics and Software Branch Chief/QD34 has the responsibility for allocating sufficient personnel resources to enable the implementation of this SA Plan. SA manpower estimates are identified in Table 3-1. Software Assurance will provide estimates of required manpower annually in support of the Planning, Programming, Budgeting, and Execution (PPBE) process.

Table 3-1. Software Assurance Manpower Estimates

SA Discipline	FY16	FY17	FY18	FY19	FY20
Management	1	1	1	1	1
Quality/V&V	5	5	5	5	5
Safety	5	5	5	5	5
Reliability	1	1	1	1	1
Total	12	12	12	12	12

3.3 Software Assurance Tasks

Software Assurance consists of the following disciplines:

- Software Quality
 - Software Quality Assurance
 - Software Quality Control
 - Software Quality Engineering
- Software Safety
- Software Reliability
- Software Verification and Validation
- Independent Verification and Validation

Each assurance discipline brings its own perspective to the tasks; the collective effort provides assurance of mission safety, reliability, and quality. For the SLS software development effort all of these disciplines apply based on the software classification assessment.

Section 3.3.1 – 3.3.30 establishes the Software Quality and Software Verification and Validation tasks, Section 15 references the Software Safety tasks, Section 16 establishes the Software Reliability tasks, and Section 17 references the Independent Verification and Validation to be performed throughout the development life cycle of the MSFC software developed for the SLS Program.

The scheduling of these tasks is driven by the software development schedule. Therefore, an SA task is performed in relationship to what software development activities are taking place. One or more SA tasks can be performed concurrently until a task is completed. The following tasks which require coordination and cooperation with the software development team and various management levels will be performed by SA.

3.3.1 Task: Perform Software Classification Assessment

SA determines the classification of the software using the criteria defined in the NPR 7150.2, and the process defined in QD-QE-016, Software Assurance Classification Assessment. The results of this assessment are coordinated and approved with the applicable stakeholders. An initial classification of the software upon project start will be performed and will be re-assessed and updated as required prior to each milestone review.

This assessment is documented in the Software Assurance Classification Report and provided to all applicable stakeholders.

3.3.2 Task: Perform Software Assurance Classification Assessment

SA determines the software assurance classification of the software using the process defined in QD-QE-016, Software Assurance Classification Assessment. The results of this assessment are coordinated and approved with the applicable stakeholders. An initial classification of the software assurance effort upon project start will be performed and will be re-assessed and updated as required prior to each milestone review.

This assessment is documented in the Software Assurance Classification Report and provided to all applicable stakeholders.

3.3.3 Task: Perform Independent Verification and Validation (IV&V) Assessment

SA determines if the software should be considered for IV&V using the process defined in QD-QE-016, Software Assurance Classification Assessment. The results of this assessment are documented in the Software Assurance Classification Report and provided to all applicable stakeholders.

3.3.4 Task: Perform Software Assurance Documentation Development and Maintenance

SA develops a Software Assurance Plan to address the requirements of SLSP DRD 1406SM-014 and applicable programmatic requirements for initial delivery at SLS Preliminary Design Review (PDR), updated as required, prior to each SLS milestone review thereafter.

The results of this task are documented as described in Section 12 and provided to all applicable stakeholders.

3.3.5 Task: Evaluate Deviations and Waivers

SA, if required, assists management, with requests for deviations and waivers and verifies that the deviation or waiver request is processed in accordance with the SLS-PLAN-008, Space Launch System (SLS) Program Configuration Management Plan and approved by the approving agency.

3.3.6 Task: Review and Approve Deliverable Software Products

The SLS-PLAN-073, Space Launch System Program Flight Software Development Plan identifies all deliverable software products to be developed, evaluated, and approved, including the standards to be followed. SA will assist the project in identifying the standards or guidelines to be followed in developing each deliverable software product. Section 4, Documentation, lists the deliverable software products to be evaluated and approved by SA, and describes the review and approval process to be followed.

The results of this task are captured in numerous forums, such as, Technical Interface Meetings (TIMs), Peer Reviews, 4511 Task Teams, SLSP Change Request process, where the SA representative has the opportunity to review and comment against the applicable product. The SMA Software Chief Safety Officer (CSO) is the SMA voting representative on the ES50/Flight & Ground Division System Review Board, as well as the Avionics Software Control Board, where these products are dispositioned. ES50 SRB minutes and ASCB directives document SMA recommendation.

3.3.7 Task: SA Problem Reporting and Corrective Action

The problem reporting and corrective action process describes the steps for (1) problem identification and corrective action occurring during software development to ensure early detection of actual or potential problems, (2) reporting of the problem to the proper authority, (3) analysis of the problem in order to propose corrective measures, (4) timely and complete corrective action, and (5) the recording and follow-up of each problem's status. Problems in this context include documentation errors, software errors, and noncompliance with standards and procedures. SA conducts periodic analysis of all reported problems to identify trends that may disclose generic problem areas. These analyses include the study of the causes, magnitude of impact, frequency of occurrence, and preventive measures.

3.3.8 Task: Evaluate Project Planning Process

Program planning involves software project management to develop and document plans for Software Development, CSCI and Software Test, Software Design, and Software Maintenance. Section 2 lists these plans. For documents to be developed, SA will assist in identifying the appropriate guidelines, standards, or Data Requirements Document (DRDs), and will assist with the tailoring of those guidelines, standards, or DRDs to meet the project's needs.

SA evaluates the Program to ensure it conducts the relevant activities stated in the Program plans. To ensure that these activities are performed as planned, SA will audit, as defined by QD-QE-011, Software Quality Assurance Software Audits and the processes that define the activity, and will use the planning document as the measure of whether those activities are being met.

The results of this task are documented and distributed to all applicable stakeholders by way of the Process Audit Report.

3.3.9 Task: Evaluate System Requirements Analysis Process

Requirements analysis establishes a common understanding of the customer's requirements between customer and the software project team.

SA conducts an audit, as defined by QD-QE-011, Software Quality Assurance Software Audits, to assure the following:

1. Ensure that requirements are reviewed to determine if they are feasible to implement, clearly stated, verifiable, and consistent.
2. Ensure that changes to allocated requirements, work products and activities are identified, reviewed, and tracked to closure.
3. Ensure that program personnel involved in the requirements definition and allocation process are trained in the necessary procedures and standards applicable to their area of responsibility in order to do the job correctly.
4. Verify that the prescribed processes for defining, documenting, and allocating requirements are followed and documented.
5. Confirm that a configuration management process is in place to control and manage the baseline.
6. Verify that requirements are documented, managed, controlled, and traced (preferably via a matrix).
7. Verify that the agreed upon requirements are addressed in the SRS.

The results of this task are documented and distributed to all applicable stakeholders by way of the Process Audit Report. SA's recommendation for corrective action requires management's disposition and implementation.

3.3.10 Task: Evaluate Software Requirements Process

The purpose of software requirements phase is to formulate, document and manage the software requirements baseline; respond to requests for clarification, correction or change; analyze impacts; revise the software requirements specification; and manage the software requirements analysis and change process.

SA conducts an audit, as defined by QD-QE-011, Software Quality Assurance Software Audits, to assure the following:

1. Ensure that the software requirements definition and analysis process and associated requirements reviews are conducted in accordance with the standards and procedures established by the program and as described in the SLS-PLAN-073, Space Launch System Program Flight Software Development Plan.
2. Assure that action items resulting from reviews of the software requirements analysis are resolved in accordance with these standards and procedures.

The results of this task are documented and distributed to all applicable stakeholders by way of the Process Audit Report. SA's recommendation for corrective action requires management's disposition and implementation.

3.3.11 Task: Evaluate Software Design Process

Preliminary design activity determines the overall structure of the software to be built. Based on the requirements identified in the previous phase, the software is partitioned into modules, and the function(s) of each module and relationships among these modules are defined.

A goal of detailed design is to define logically how the software will satisfy the allocated requirements. The level of detail of this design must be such that the coding of the computer program can be accomplished by someone other than the original designer. Section 4, Documentation, lists the software design documents.

SA conducts an audit, as defined by QD-QE-011, Software Quality Assurance Software Audits, to assure the following:

1. Ensure that the software design process and associated design reviews are conducted in accordance with standards and procedures established by the project and as described in SLS-PLAN-073, Space Launch System Program Flight Software Development Plan.
2. Assure that action items resulting from reviews of the design are resolved in accordance with these standards and procedures.
3. Ensure that the method used for tracking and documenting the development of a software unit is implemented and is kept current.

The results of this task are documented and distributed to all applicable stakeholders by way of the Process Audit Report. SA's recommendation for corrective action requires management's disposition and implementation.

3.3.12 Task: Evaluate Coding and Unit Testing Process

Software implementation or coding is the point in the software development cycle at which the design is finally implemented. The process includes unit testing of the software code.

SA conducts an audit, as defined by QD-QE-011, Software Quality Assurance Software Audits, to assure the following:

1. That the coding process, associated code reviews, and software unit testing are conducted in conformance with the standards and procedures established as described in the SDP.
2. That action items resulting from reviews of the code are resolved in accordance with these standards and procedures.
3. Ensure that the method used for tracking and documenting the development of a software unit is implemented and is kept current.

The results of this task are documented and distributed to all applicable stakeholders by way of the Process Audit Report. SA's recommendation for corrective action requires management's disposition and implementation.

3.3.13 Task: Evaluate Unit Integration and Testing, CSCI Qualification Testing, Integration and Testing, and System Qualification Testing

Software integration and test activities combine individually developed components together in the developing environment to ensure that they work together to complete the software and system functionality. For joint hardware and software development, integration requires close synchronization of hardware and software to meet designated integration and test milestones.

In the integration and test phase of the development lifecycle, the testing focus shifts from an individual component correctness to the proper operation of interfaces between components, the flow of information through the system, and the satisfaction of system requirements.

SA conducts an audit, as defined by QD-QE-011, Software Quality Assurance Software Audits, to assure the following:

1. That the software test activities are identified, test environments have been defined, and guidelines for testing have been designed. SA will verify the software integration process, software integration testing activities and the software performance testing activities are being performed in accordance with the SLS-PLAN-073, Space Launch System Program Flight Software Development Plan, the software design, the plan for software testing, and established software standards and procedures.
2. That as many of the software integration tests as necessary and all of the software performance tests are witnessed to ensure that the approved test procedures are being followed, that accurate records of test results are being kept, that all discrepancies discovered during the tests are being properly reported, that test results are being analyzed, and the associated test reports are completed.
3. That discrepancies discovered during software integration and performance tests are identified, analyzed, and corrected; software unit tests, and software integration tests are re-executed as necessary to validate corrections made to the code; and the software unit's

design, code, and test is updated based on the results of software integration testing, software performance testing, and corrective action process.

4. That the responsibility for testing and for reporting on results has been assigned to a specific organizational element.
5. That the software is tested.
6. That test reports are developed.

The results of this task are documented and distributed to all applicable stakeholders by way of the Process Audit Report. SA's recommendation for corrective action requires management's disposition and implementation.

3.3.14 Task: Non-Deliverable Software

The program may use non-deliverable software in the development of deliverable software as long as the operation and support of the deliverable software after delivery does not depend on the non-deliverable software or provision is made to ensure that the acquirer has or can obtain the same software. SA ensures that the use of non-deliverable software meets the above criteria, that is, deliverable software is not dependent on non-deliverable software to execute, or verify that the acquirer can obtain the same software. SA conducts an audit, as defined by QD-QE-011, Software Quality Assurance Software Audits, to ensure that all non-deliverable software used performs its intended functions.

The results of this task are documented and distributed to all applicable stakeholders by way of the Process Audit Report. SA's recommendation for corrective action requires management's disposition and implementation.

3.3.15 Task: Evaluate Non-Developmental Software

Non-Developmental Software (NDS) is software that is provided by the contractor, the Government, or a third party. NDS may be referred to as reusable software, Government-furnished software, or commercially available software depending on it source. SA conducts an audit, as defined by QD-QE-011, Software Quality Assurance Software Audits, to ensure that non-developmental software performs its intended functions as defined by the SRS.

The results of this task are documented and distributed to all applicable stakeholders by way of the Process Audit Report. SA's recommendation for corrective action requires management's disposition and implementation.

3.3.16 Task: Verify Software Configuration Audits

SA conducts an audit, as defined by QD-QE-011, Software Quality Assurance Software Audits, to ensure that a formal software configuration audit in accordance with applicable documentation is conducted. A software configuration audit is a formal examination of a CSCI. SA ensures that the Functional Configuration Audit (FCA) and/or the Physical Configuration Audit (PCA) as

detailed in the SLS-PLAN-074, Space Launch System Program Flight Software Configuration Management Plan are conducted prior to software release.

The results of this task are documented and distributed to all applicable stakeholders by way of the Process Audit Report. SA's recommendation for corrective action requires management's disposition and implementation.

3.3.17 Task: Verify Implementation of Requirements Management

The purpose of Requirements Management is to establish a common understanding between the customer and the software project of the customer's requirements that will be addressed by the software project.

SA reviews and/or audits the activities and work products for managing the allocated requirements and reports the results. This audit is conducted as defined by QD-QE-011, Software Quality Assurance Software Audits. The project software assurance lead is responsible for scheduling this audit every fiscal year.

The results of this task are documented and distributed to all applicable stakeholders by way of the Process Audit Report.

3.3.18 Task: Verify Implementation of Software Project Planning

The purpose of Software Project Planning is to establish reasonable plans for performing the software engineering and for managing the software project.

SA reviews and/or audits the activities and work products for software project planning and reports the results. This audit is conducted as defined by QD-QE-011, Software Quality Assurance Software Audits. The project software assurance lead is responsible for scheduling this audit every fiscal year.

The results of this task are documented and distributed to all applicable stakeholders by way of the Process Audit Report.

3.3.19 Task: Verify Implementation of Software Project Monitoring and Control

The purpose of Software Project Monitoring and Control is to establish adequate visibility into actual progress so that management can take effective actions when the software project's performance deviates significantly from the software plans.

SA reviews and/or audits the activities and work products for software project monitoring and control and reports the results. This audit is conducted as defined by QD-QE-011, Software Quality Assurance Software Audits. The project software assurance lead is responsible for scheduling this audit every fiscal year.

The results of this task are documented and distributed to all applicable stakeholders by way of the Process Audit Report.

3.3.20 Task: Verify Implementation of Software Supplier Agreement Management

The purpose of Software Supplier Agreement Management is to select qualified software subcontractors and manage them effectively.

SA reviews and/or audits the activities and work products for software supplier agreement management and reports the results. This audit is conducted as defined by QD-QE-011, Software Quality Assurance Software Audits. The project software assurance lead is responsible for scheduling this audit every fiscal year.

The results of this task are documented and distributed to all applicable stakeholders by way of the Process Audit Report.

3.3.21 Task: Verify Implementation of Measurement and Analysis

The purpose of Measurement and Analysis is to develop and sustain a measurement capability that is used to support management information needs.

SA reviews and/or audits the activities and work products for measurement and analysis and reports the results. This audit is conducted as defined by QD-QE-011, Software Quality Assurance Software Audits. The project software assurance lead is responsible for scheduling this audit every fiscal year.

The results of this task are documented and distributed to all applicable stakeholders by way of the Process Audit Report.

3.3.22 Task: Verify Implementation of Software Configuration Management

The purpose of Software Configuration Management is to establish and maintain the integrity of the products of the software project throughout the project's software life cycle.

SA reviews and/or audits the activities and work products for software configuration management and reports the results. This audit is conducted as defined by QD-QE-011, Software Quality Assurance Software Audits. The project software assurance lead is responsible for scheduling this audit every fiscal year.

The results of this task are documented and distributed to all applicable stakeholders by way of the Process Audit Report.

3.3.23 Task: Verify Implementation of Requirements Development

The purpose of Requirements Development is to produce and analyze customer, product, and product-component requirements.

SA reviews and/or audits the activities and work products for requirements development and reports the results. This audit is conducted as defined by QD-QE-011, Software Quality Assurance Software Audits. The project software assurance lead is responsible for scheduling this audit every fiscal year.

The results of this task are documented and distributed to all applicable stakeholders by way of the Process Audit Report.

3.3.24 Task: Verify Implementation of Technical Solution

The purpose of Technical Solution is to design, develop, and implement solutions to requirements. Solutions, designs, and implementations encompass products, product components, product related lifecycle processes either singly or in combination as appropriate.

SA reviews and/or audits the activities and work products for technical solution and reports the results. This audit is conducted as defined by QD-QE-011, Software Quality Assurance Software Audits. The project software assurance lead is responsible for scheduling this audit every fiscal year.

The results of this task are documented and distributed to all applicable stakeholders by way of the Process Audit Report.

3.3.25 Task: Verify Implementation of Product Integration

The purpose of Product Integration is to assemble the product from the product components, ensure that the product, as integrated, functions properly, and deliver the product.

SA reviews and/or audits the activities and work products for product integration and reports the results. This audit is conducted as defined by QD-QE-011, Software Quality Assurance Software Audits. The project software assurance lead is responsible for scheduling this audit every fiscal year.

The results of this task are documented and distributed to all applicable stakeholders by way of the Process Audit Report.

3.3.26 Task: Verify Implementation of Verification

The purpose of Verification is to ensure that selected work products meet their specified requirements.

SA reviews and/or audits the activities and work products for verification and reports the results. This audit is conducted as defined by QD-QE-011, Software Quality Assurance Software Audits. The project software assurance lead is responsible for scheduling this audit every fiscal year.

The results of this task are documented and distributed to all applicable stakeholders by way of the Process Audit Report.

3.3.27 Task: Verify Implementation of Validation

The purpose of Validation is to demonstrate that a product or product component fulfills its intended use when placed in its intended environment.

SA reviews and/or audits the activities and work products for validation and reports the results. This audit is conducted as defined by QD-QE-011, Software Quality Assurance Software Audits. The project software assurance lead is responsible for scheduling this audit every fiscal year.

The results of this task are documented and distributed to all applicable stakeholders by way of the Process Audit Report.

3.3.28 Task: Verify Implementation of Integrated Project Management

The purpose of Integrated Project Management is to establish and manage the project and the involvement of the relevant stakeholders according to an integrated and defined process that is tailored from the organization's set of standard processes.

SA reviews and/or audits the activities and work products for integrated project management and reports the results. This audit is conducted as defined by QD-QE-011, Software Quality Assurance Software Audits. The project software assurance lead is responsible for scheduling this audit every fiscal year.

The results of this task are documented and distributed to all applicable stakeholders by way of the Process Audit Report.

3.3.29 Task: Verify Implementation of Risk Management

The purpose of Risk Management is to identify potential problems before they occur so that risk mitigating activities can be planned and invoked as needed across the life of the product to mitigate adverse impacts on achieving objectives.

SA reviews and/or audits the activities and work products for risk management as defined in SLS-PLAN-180, Space Launch System Program Risk and Opportunity Management Plan and reports the results. This audit is conducted as defined by QD-QE-011, Software Quality Assurance Software Audits. The project software assurance lead is responsible for scheduling this audit every fiscal year.

The results of this task are documented and distributed to all applicable stakeholders by way of the Process Audit Report.

3.3.30 Task: Verify Implementation of Decision Analysis and Resolution

The purpose of Decision Analysis and Resolution is to analyze possible decisions using a formal evaluation process that evaluates identified alternatives against established criteria.

SA reviews and/or audits the activities and work products for decision analysis and resolution and reports the results. This audit is conducted as defined by QD-QE-011, Software Quality Assurance Software Audits. The project software assurance lead is responsible for scheduling this audit every fiscal year.

The results of this task are documented and distributed to all applicable stakeholders by way of the Process Audit Report.

3.4 Stakeholder Responsibilities

The ultimate responsibility for the quality of the SLS flight software and associated products produced by the development organization rests with the Software Project Lead. SA is responsible to the SLS Integrated Avionics and Software Discipline Lead Engineer (DLE) and for implementation of the SA procedures defined in this plan. Figure 3-1 depicts the SLS, Software Development Organization, and Software Assurance structure.

SA monitors activities and review products for compliance to applicable standards, procedures, and the SLS-PLAN-073, Space Launch System Program Flight Software Development Plan. The results of SA monitoring and analysis along with SA's recommendations for corrective action are reported to the SLS Integrated Avionics and Software Discipline Lead Engineer (DLE), and relevant stakeholders. All documents and software approved by the Software Project Lead for release have been reviewed and approved by SA. Table 3-2 is a responsibility matrix for the tasks identified in Section 3.3, SA Tasks.

Table 3-2. Stakeholder Responsibility Matrix

	QD34/Branch Chief	SA Rep.	Integrated A/S DLE	Software Project Lead	FSW Team Leads
3.3.1 Task: Perform Software Classification Assessment					
Assessment		●			
Approval	●	●	●	●	
Report		●			
3.3.2 Task: Perform Software Assurance Classification Assessment					
Assessment		●			
Approval	●	●	●	●	
Report		●			
3.3.3 Task: Perform Independent Verification and Validation Assessment					
Assessment		●			
Approval	●	●	●	●	
Report		●			
3.3.4 Task: Perform Software Assurance Documentation Development and Maintenance					
Develop		●			
Review	●	●	●	●	●
Rework		●			
Approve	●	●	●	●	
3.3.5 Task: Evaluate Deviations and Waivers					
Document				●	●

	QD34/Branch Chief	SA Rep.	Integrated A/S DLE	Software Project Lead	FSW Team Leads
Recommendation		●		●	
Approval			●		
Report	●	●			
3.3.6 Task: Review and Approve Deliverable Software Products					
Evaluate		●		●	●
Report		●			
Rework					●
Approval		●	●	●	
3.3.7 Task: SA Problem Reporting and Corrective Action					
Document		●		●	●
Recommendation		●		●	●
Approval		●		●	●
Report					●
3.3.8 Task: Evaluate Project Planning and Oversight Process					
Evaluate Process		●			
Resolve Issues	●	●	●	●	●
Report		●			
3.3.9 Task: Evaluate System Requirements Analysis Process					
Evaluate Process		●			
Resolve Issues	●	●	●	●	●
Report		●			
3.3.10 Task: Evaluate Software Requirements Process					
Evaluate Process		●			
Resolve Issues	●	●	●	●	●
Report		●			
3.3.11 Task: Evaluate Software Design Process					
Evaluate Process		●			
Resolve Issues	●	●	●	●	●
Report		●			
3.3.12 Task: Evaluate Coding and Unit Testing Process					
Evaluate Process		●			
Resolve Issues	●	●	●	●	●

	QD34/Branch Chief	SA Rep.	Integrated A/S DLE	Software Project Lead	FSW Team Leads
Report		●			
3.3.13 Task: Evaluate Unit Integration and Testing, CSCI Qualification, HW/SW Integration and Testing, System Qualification Testing					
Evaluate Process		●			
Resolve Issues	●	●	●	●	●
Report		●			
3.3.14 Task: Non-Deliverable Software					
Evaluate Process		●			
Resolve Issues	●	●	●	●	●
Report		●			
3.3.15 Task: Evaluate Non-Development Software					
Evaluate Process		●			
Resolve Issues	●	●	●	●	●
Report		●			
3.3.16 Task: Verify Software Configuration Audits					
Evaluate Process		●			
Resolve Issues	●	●	●	●	●
Report		●			
3.3.17 Task: Verify Implementation of Requirements Management **3.3.18 Task: Verify Implementation of Software Project Planning** **3.3.19Task: Verify Implementation of Software Project Monitoring and Control** **3.3.20 Task: Verify Implementation of Software Supplier Agreement Management** **3.3.21 Task: Verify Implementation of Measurement and Analysis** **3.3.22 Task: Verify Implementation of Software Configuration Management** **3.3.23 Task: Verify Implementation of Requirements Development** **3.3.24 Task: Verify Implementation of Technical Solution** **3.3.25 Task: Verify Implementation of Product Integration** **3.3.26 Task: Verify Implementation of Verification** **3.3.27 Task: Verify Implementation of Validation** **3.3.28 Task: Verify Implementation of Integrated Project Management** **3.3.29 Task: Verify Implementation of Risk Management** **3.3.30 Task: Verify Implementation of Decision Analysis and Resolution**					
Evaluate Process		●			
Resolve Issues	●	●		●	●
Report		●			

3.5 Schedule

The Software Assurance schedule is closely coordinated with the software development schedule in the SLS-PLAN-073, Space Launch System Program Flight Software Development Plan. The project software assurance lead will generate each fiscal year a Process Audit schedule covering those audits identified in Sections 3.3.17 through 3.3.30.

4.0 DOCUMENTATION

The documentation which specifies, describes, and supports the software development process will be created and updated throughout the development cycle. The SLS-PLAN-073, Space Launch System Program Flight Software Development Plan identifies the software deliverable products. For convenience they are also listed in Table 4-1. SA has identified the applicable standard to which these software products must comply with. Any tailoring guidelines used will be defined in the software product being developed. Software Assurance Review/Approval of Technical Documents (QD-QE-009) will be used for the review/approval of technical documents (and changes therein) identified in Table 4-1.

The SLS-PLAN-073, Space Launch System Program Flight Software Development Plan identifies the non-deliverable products. For convenience they are also listed in Table 4-2.

Table 4-1. Deliverable Software Products

Deliverable Products	Standard
Software Development Plan (SDP)	MSFC DRD STD/SW-SDP
Software Configuration Management Plan (SCMP)	MSFC DRD STD/SW-SCMP
Software Verification and Validation Plan	N/A
User's Manual	MSFC DRD STD/SW-SUM
Data Dictionary (Appendices to SDD)	MSFC DRD STD/SW-SWDD
Software Maintenance Plan (SMP)	MSFC DRD STD/SW-SMP
Software Requirements Specification (SRS)	MSFC DRD STD/SW-SRS
Software Test Procedure (STPr)	MSFC DRD STD/SW-STPr
Software Test Report (STR)	MSFC DRD STD/SW-STR
Software Version Description (SVD)	MSFC DRD STD/SW-SVD
Software Design Description (SDD)	MSFC DRD STD/SW-SDD
Software Code	Executable code, no standard
FSW Acceptance Data Package (ADP)	SLS Program Criteria

Table 4-2. Non-Deliverable Products

NON-Deliverable Products	Standard
Software Metrics Report (SMR)	MSFC DRD STD/SW-SMR
Software Inspection/Peer Review Report (SIPRR)	MSFC DRD STD/SW-SIPRR

5.0 STANDARD, PRACTICES, CONVENTIONS, AND METRICS

5.1 Participants in Software Assurance

The SLS-PLAN-073, Space Launch System Program Flight Software Development Plan defines the procedures by which the software development staff ensures the quality of the products throughout the life-cycle.

5.2 Metrics

The following SA measurements are collected during the course of each software development effort to monitor the status of the SA tasks:

Process Audits Planned versus Actual

SA is responsible for reporting these measurements as described in Section 12.1.4.

6.0 REVIEWS

6.1 Technical Reviews

A primary component of engineering quality into software is the conduct of technical reviews of software products, both deliverable and non-deliverable. Participants of a technical review include persons with technical knowledge of the software products to be reviewed. The purpose of the technical review will be to focus on in-progress and final software products rather than the materials generated especially for the review. SA will assure that technical reviews are accomplished and will selectively attend them in accordance with approved sampling techniques as defined in QD-QE-10 Software Assurance Software Milestone Review Support. SLS-PLAN-017, Space Launch System (SLS) Program Review Plan establishes and defines these reviews.

Technical reviews will be conducted to review evolving software products, demonstrate proposed technical solutions, and provide insight and obtain feedback on the technical effort.

6.2 Management Reviews

Each month as part of the ES50 Flight software (FSW) Monthly Review, Software Assurance presents to various levels of management, process audit schedule progress and process audit open SCR aging metric.

Additionally, Software Assurance presents monthly to the SMA Director, as part of the overall SMA Monthly Status Review, significant accomplishments and problems associated with the SA Program.

6.3 Peer Reviews

Software Assurance participates in the FSW MSFC-developed peer reviews as defined in SLS-PLAN-073, Space Launch System Program Flight Software Development Plan. Typical products subject to peer review include but not limited to planning documentation, software requirements specifications, design documentation, and source code.

Peer reviews are conducted as defined in the organization work instruction ES50-SOP09, Peer Review.

7.0 TEST

Software Assurance involvement in testing begins early in the requirements phase by assuring that all software requirements are verifiable by one or more of the acceptable verification methods. The verification methods include; inspection, test, analysis, or demonstration.

During the software implementation and unit test software lifecycle phase, Software Assurance will ensure that the coding process and software unit testing is conducted and complies with the standards and procedures established in the SLS-PLAN-073, Space Launch System Program Flight Software Development Plan.

In the test phase of the software lifecycle, Software Assurance will support formal software testing as defined in QD-QE-012, Software Quality Assurance Support of Formal Software Testing.

8.0 SA PROBLEM REPORTING AND RESOLUTION

This section describes the reporting and control system used by SA to record discrepancies and to monitor the implementation of corrective action. SA strives to resolve any concern/issue at the lowest level possible; however, when an adequate resolution cannot be achieved SA will raise the concern/issue up through management until a resolution is reached or acceptance of the concern/issue by the SLS Integrated Avionics and Software Discipline Lead Engineer (DLE). Concerns/issues may be identified during a product audit, process audit, peer review, or a milestone review. The documenting and tracking of identified concerns/issues is dependent on how the concern/issue is identified by SA.

8.1 Product Audit Reporting

During the review of software deliverable documentation (i.e. Software Development Plan), SA documents concerns/issues and associated corrective action needed to resolve SA

concerns/issues utilizing the appropriate mechanism provided by the software developing organization responsible for conducting the review and forwards this to the applicable stakeholders for consideration and resolution.

8.1.1 Product Audit Corrective Action Implementation

SA will re-assess affected software product for resolution of earlier submitted concerns/issues at the next release of the software product and/or prior to System Review Board (SRB) disposition of the software product.

8.1.2 Product Audit Unresolved Concerns/Issues Escalation

For those concerns/issues not addressed by the documentation developer, SA will notify the Software Project Team Lead of any outstanding concerns/issues not addressed. If an adequate resolution still cannot be reached, SA will notify the Software Project Lead and QD34 Branch Chief of any outstanding concerns/issues. Concerns/issues which cannot be resolved will be escalated to the ES50 Division Manager and the SMA Vehicle Systems Department Manager. Unresolved concerns/issues will be escalated to the SLS Integrated Avionics and Software DLE and SMA Avionics and Software Chief Safety Officer. Figure 8-1 shows this escalation.

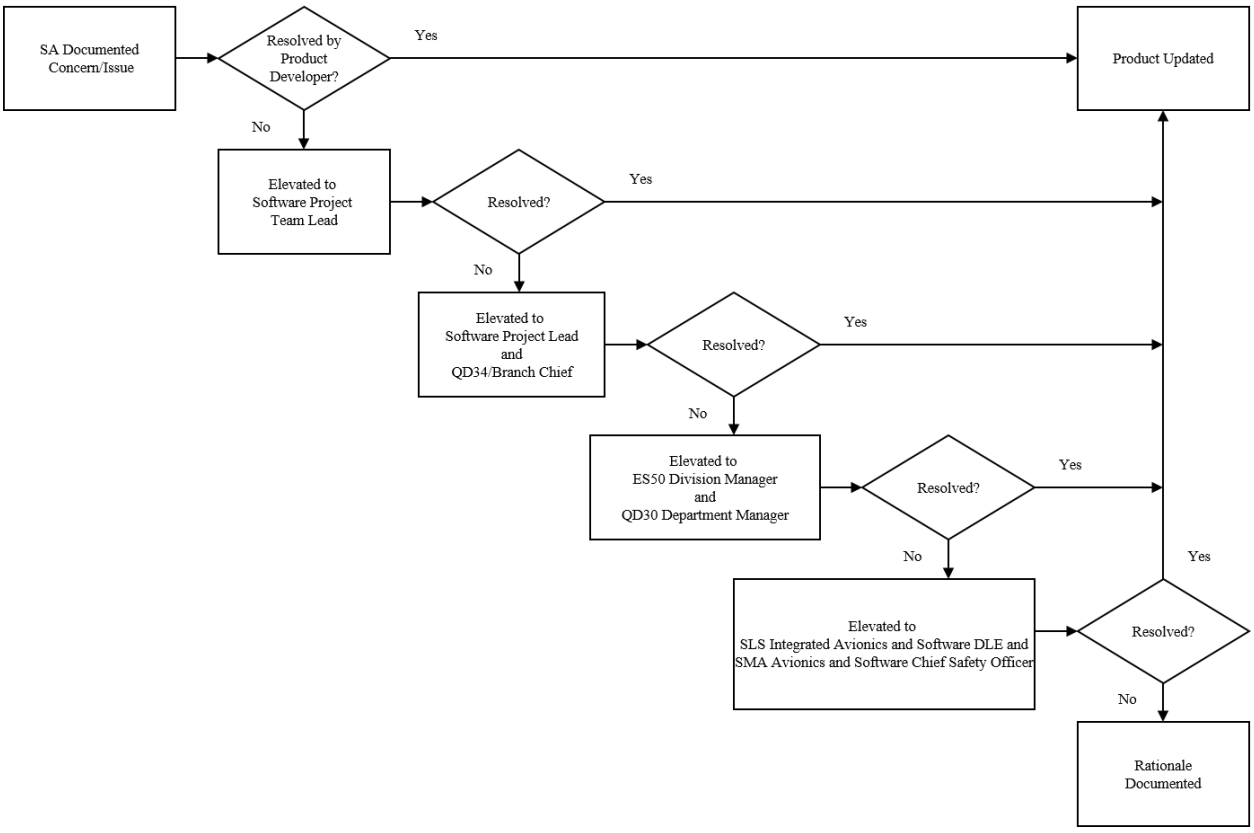


Figure 8-1. Escalation of SA Product Audit Unresolved Concern/Issue

8.2 Process Audit Reporting

At the conclusion of a process audit, a draft process audit report is produced and presented during the exit briefing to the Software Project Lead and audit interviewees. Any documented audit findings/observations along with suggested corrective action is discussed, allowing the audited organization to present additional information which may address these documented findings/observations. A final process audit report is produced documenting any unresolved audit findings/observations from the exit briefing and distributed to the relevant stakeholders. Each process audit finding/observation is entered into the SA database for tracking and trending purposes.

8.2.1 Process Audit Findings/Observations Corrective Action Implementation

Depending on the corrective action SA will ensure associated work products are updated and/or conduct a follow-up process audit to ensure process implementation.

8.2.2 Process Audit Unresolved Findings Escalation

For those findings not addressed by the Software Project Lead, SA will notify the Software Project Lead and QD34 Branch Chief of any outstanding findings. Findings which cannot be resolved will be escalated to the ES50 Division Manager and the SMA Vehicle Systems Department Manager. Unresolved concerns/issues will be escalated to the SLS Integrated Avionics and Software DLE and SMA Avionics and Software Chief Safety Officer. Figure 8-2 shows this escalation.

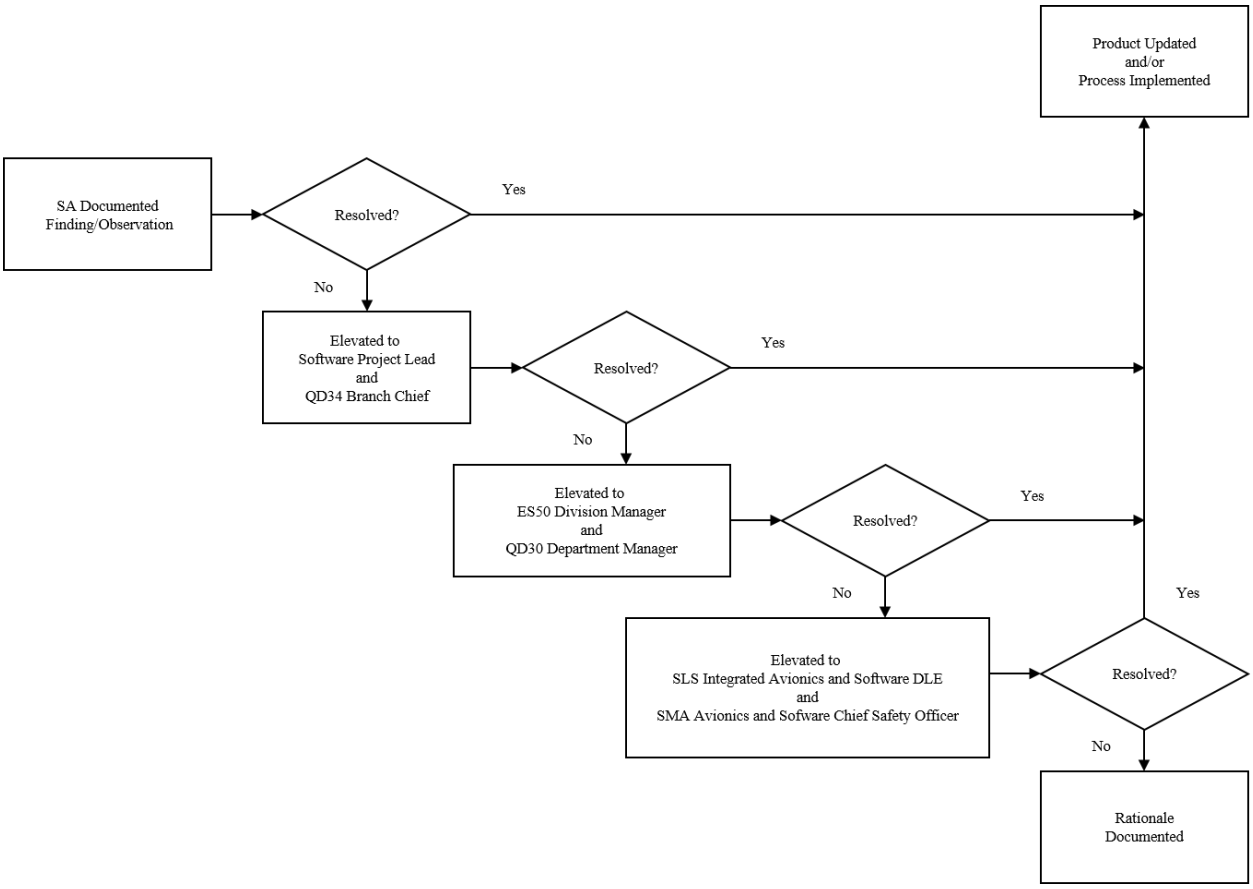


Figure 8-2. Escalation of SA Process Audit Unresolved Finding/Observation

8.3 Peer Review Reporting

SA will document all defects/actions/typos and suggested corrective action, utilizing the appropriate mechanism provided by the software developing organization responsible for conducting the peer review. SA will participate in the peer review discussion meeting(s) to ensure SA documented defects/actions/typos and suggested corrective action are adequately addressed.

8.3.1 Peer Review Defects/Typos Corrective Action Implementation

SA will re-assess the peer review software product for resolution of earlier submitted defects/typos prior to System Review Board (SRB) disposition of the software product.

8.3.2 Peer Review Unresolved Defect Escalation

For those defects not addressed by the documentation developer, SA will notify the Software Project Team Lead of any outstanding concerns/issues not addressed. If an adequate resolution

still cannot be reached, SA will notify the Software Project Lead and QD34 Branch Chief of any outstanding concerns/issues. Concerns/issues which cannot be resolved will be escalated to the ES50 Division Manager and the SMA Vehicle Systems Department Manager. Unresolved concerns/issues will be escalated to the SLS Integrated Avionics and Software DLE and SMA Avionics and Software Chief Safety Officer. Figure 8-3 shows this escalation.

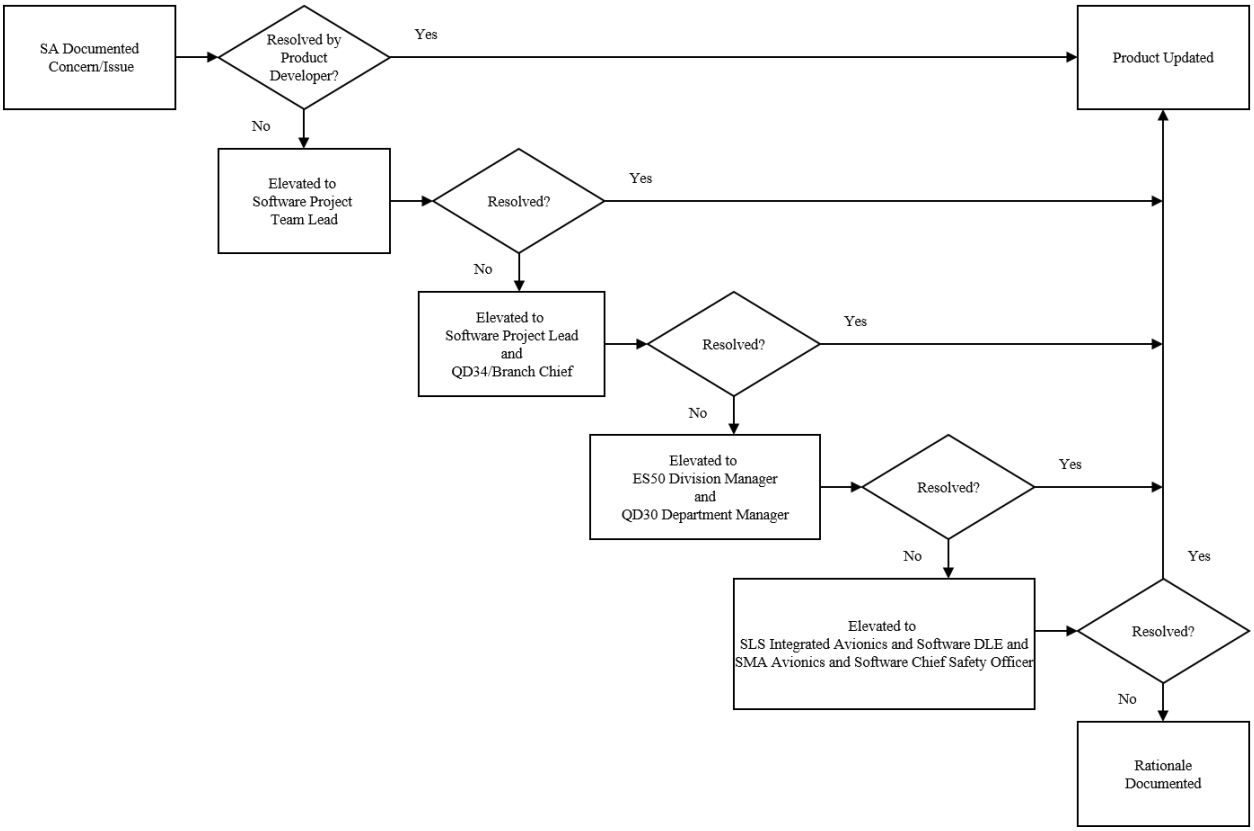


Figure 8-3. Escalation of SA Peer Review Unresolved Defect

8.4 Milestone Review Reporting

In support of the SLS milestone reviews, SA will document review item discrepancy (RID) per the process defined in the applicable milestone review plan.

8.4.1 Milestone Review RID Corrective Action Implementation

As the RID initiator SA, concurrence is needed from SA for RID closure.

8.4.2 Milestone Review RID Escalation

SA will follow the process defined in the applicable SLS milestone review plan to elevate any issues associated with the submitted RID.

9.0 TOOLS, TECHNIQUES, AND METHODOLOGIES

9.1 Tools

Table 9-1 identifies the tools available to SA to support the completion of the tasks identified within this plan.

Table 9-1. SA Tools

Nomenclature	Type	Purpose
Microsoft Word	Application	Word processor
Microsoft Excel	Application	Spreadsheet
Microsoft PowerPoint	Application	Presentation graphics
Microsoft Outlook	Application	Personal information manager and communications
Microsoft Project 2003	Application	Scheduling
Automated Requirement Measurement (ARM) Tool	Application	Provides measures that can be used to assess the quality attributes of a software requirement specification (SRS).
SBM©	Application	Database of inputting and tracking SA audit findings.
SA Classification Report	Report	Documents the Software Classification Assessment; SA Effort/Priority; Software Safety Assessment; IV&V Assessment
SW Classification Score Sheet Template	Excel Spreadsheet	Results used to determine the level of SA to be applied.
Software Configuration Management Plan Checklist	Checklist	Documents Product Audit Objective Evidence
Software Change Reporting Process Checklist	Checklist	Documents Process Audit Objective Evidence
Software Requirements Specification Checklist	Checklist	Documents Product Audit Objective Evidence
Software CDR Checklist	Checklist	Documents Process Audit Objective Evidence
Software PDR Checklist	Checklist	Documents Process Audit Objective Evidence
Software Risk Management Plan Checklist	Checklist	Documents Product Audit Objective Evidence
Software Requirements Review Checklist	Checklist	Documents Process Audit Objective Evidence
Software Requirements Trace Matrix Checklist	Checklist	Documents Process Audit Objective Evidence
Software Test Plan Checklist	Checklist	Documents Product Audit Objective Evidence
Software Test Readiness Review Checklist	Checklist	Documents Process Audit Objective Evidence
Adobe Reader©	Application	File conversion for record integrity
DOORS©	Application	Provides traceability/compliance of SA requirements
SoftRel©	Excel Spreadsheet	SoftRel© Full-scale Survey, Software Reliability Prediction and Growth Models

9.2 Techniques

SA applies the following techniques as part of conducting their tasks:

- **Requirements Analysis** – As conducted by SA ensures that all software requirements are: unambiguous, complete, consistent, verifiable, and traceable.
- **Audits** – Are used by SA to examine the conformance of a development process to procedures and the conformance of products to standards. An SA audit also can examine the conformance of the actual status of the development activity to the reported status.
- **Bi-Directional Traceability** – Aids SA in determining that all source requirements have been completely addressed and that all lower level requirements can be traced to a valid source requirement.

9.3 Methodologies

Methodologies are an integrated set of the above tools and techniques. The methodologies should be well documented for accomplishing the task and provide a description of the process to be used. The following methodologies are briefly discussed here, with more detail provided for each in its applicable SMA organizational issuances used by SA in performing their tasks:

- **Product Audits** – Software Assurance using the appropriate checklist and associated programmatic documentation, assesses the software product under review for compliance to the applicable Data Requirement Description (DRD) and project requirements. SMA Organizational Instruction QD-QE-009, Software Assurance Review/Approval of Technical Documents details this methodology.
- **Process Audits** – Software Assurance Process audits has four phases: 1). Planning and preparation, 2). Site visit, 3). Reporting, and 4). Follow-up. Software Assurance tailors applicable audit checklist and identifies key software development personnel for interviews. Software Assurance conducts interviews and assesses objective evidence for compliance to the Software Development Plan and applicable organizational instructions. Software Assurance conducts exit briefing with the software development organization and then publishes the final audit report. If warranted, a follow-up audit may be conducted. SMA Organizational Instruction QD-QE-011, Software Quality Assurance Software Audits details this methodology.
- **Milestone Reviews** – Software Assurance supports the SLS Program, in participating in software milestone reviews by evaluating milestone deliverables for compliance with programmatic requirements. Prepare documentation issues for each nonconformance/discrepancy identified and track to closure. SMA Organizational Instruction QD-QE-010, Software Assurance Milestone Review Support details this methodology.
- **Testing** – Software Assurance will witness formal testing of all flight software. SA will review test documentation in accordance with QD-QE-009, Software Assurance

Review/Approval of Technical Documents. During testing SA will ensure approved test procedures are being used and that the test environment is in accordance with approved test documentation. All test non-conformances are documented and tracked to closure. SA will ensure all test reports are generated upon conclusion and conform to the applicable standards. SMA Organizational Instruction QD-QE-012, Software Quality Assurance Support of Formal Testing details this methodology.

10.0 MEDIA CONTROL

The software control methods and facilities are described in SLS-PLAN-073, Space Launch System Program Flight Software Development Plan. Software Assurance will conduct evaluations of the software control process to verify that the process of controlling the software is effective and in compliance with the SLS-PLAN-073, Space Launch System Program Flight Software Development Plan.

11.0 SUPPLIER CONTROL

Prior to any purchase of software to support the development effort, SA and project members will define and provide complete requirements to the supplier/vendor. Part of the evaluation process will require the supplier or vendor to describe their technical support, handling of user questions and problems, and software product upgrades.

12.0 RECORDS, COLLECTION, MAINTENANCE, AND RETENTION

12.1 Records

SA tasks are documented by records, analyses, and reports that provide a history of product quality and reliability throughout the software life cycle. The type of SA records identified in Table 12-1 will be developed by SA throughout the software development effort of the SLS life-cycle. The SA lead for the SLS Program will be the custodian for all records and the SA repository for all records is identified in Table 12-1, unless otherwise noted. A brief description of each record is provided below.

12.1.1 Software Assurance Plan

This Software Assurance Plan is the master planning and control document of SA tasks to be applied to the software development effort.

12.1.2 Process Audit Reports

The objective of the process audit report is to present a clear picture, that a particular development process is effectively producing a quality product, to program management. The report is to be clear, objective, and factual.

12.1.3 Assessments

- SA conducts and/or assures that several types of assessments are performed and documented. A brief description for each assessment is provided below. **Software Classification Assessment** – An assessment to evaluate the characteristics of the software in determining the software classification and level of control. This assessment is documented on the Software Assurance Classification Report.
- **Software Assurance Classification Assessment** – An assessment to evaluate the software classification to determine the priority and level of effort of the software assurance to be applied to the software development effort. This assessment is documented on the Software Assurance Classification Report.
- **Independent Verification and Validation Assessment** – An assessment of the characteristics of the software, software criticality and software classification in determining if IV&V is required for the software development effort. This assessment is documented on the Software Assurance Classification Report.

12.1.4 Status Reports

Status reports are provided/presented monthly to management, each report addresses the following:

- FSW SA Process Audit Schedule Progress
- ALL SLS FSW SA CRs by Process Area
- SLS SA FSW Process Audit OPEN CRs Aging Metric
- SLS SA FSW Issues/Concerns, as applicable

12.1.5 Software Reliability Analysis

The following Software Reliability Analyses are be performed across the software development cycle with analysis results reported at each milestone review in accordance with Section 16.1.2. Updating will continue throughout the software development lifecycle:

- Software Failure Modes and Effects Analysis (FMEA)
- Software Reliability Prediction Analysis
- Software Reliability Growth Analysis

Table 12-1. Software Assurance Developed Records

Record Type	SA Repository Location
SLS FSW Software Assurance Plan	Maintained by ES50 SRB
Process Audit Report(s)	[server]:***\SA\ProcessAudits
Assessment(s)	[server]:\ ***\SA\Assessments
Status Report(s)	[server]:\ ***\SA\MonthyStatus
Software Reliability Analysis	[server]:\ ***\SA\SoftwareReliabilityAnalysis

*** – Applicable Organizational Acronym

12.2 Collection

The SA lead for the SLS FSW is responsible for depositing all records in the appropriate SA repository location within a reasonable timeframe of record completion. To maintain SA record integrity conversion to Adobe Reader format may be used or "read only" attribute will be assigned to the record file prior to deposit in the SA repository location.

12.3 Distribution

Table 12-2 represents the applicable stakeholders to which the records identified in Table 12-1 are distributed.

Table 12-2. Distribution of SA Records

Record	Center IV&V Liaison	QD34 Branch Chief	Integrated A/S DLE	Software Project Lead	FSW Team Leads
SAP		●	●	●	●
Process Audit Reports		●	●	●	●
Assessments	*●	●	●	●	
Status Reports		●	●	●	
Software Reliability Analyses		●	●	●	●

*Only the Software Classification, Software Assurance Classification, and IV&V Assessments.

12.4 Maintenance

This plan is the only SA developed record which requires maintenance. Section 1.3 defines the configuration control process followed by SA in the maintenance of this plan.

SA Audit Reports, Assessments, Status Reports and Software Reliability Analyses are completed reports/analyses, thus requiring no maintenance. To maintain SA record integrity conversion to Adobe Reader format will be used on all completed records prior to deposit in the SA repository location.

Checklists are developed as generic and are tailorable for any given project. The SA auditor is responsible for tailoring these checklists as defined in Organizational Issuance QD-QE-011, Software Quality Assurance Software Audits.

12.5 Retention

All SA records will be collected and maintained as described in paragraphs 12.2 and 12.4 in the SA repository or archival storage for 5 years after retirement of the SLS program.

13.0 TRAINING

Trained personnel will be provided to perform SA tasks identified within this plan. Training will be accomplished to maintain those skills. In some cases, training will be accomplished using On-the-Job (OJT) training.

SLS Program specific training is identified in Table 13-1.

Table 13-1. SLS Program Specific SA Training

Training	Skill
DOORS©	Requirements Traceability Tool
SoftRel©	Software Reliability Analysis Tool

14.0 RISK MANAGEMENT

The SLS Program has developed SLS-PLAN-180, Space Launch System (SLS) Program Risk and Opportunity Management Plan to address level risks. Additionally, the software development organization has defined a risk management process in SLS-PLAN-073, Space Launch System Program Flight Software Development Plan, to address project risks which may become elevated to a Program risk. SA will review and evaluate the technical risk analysis and any risk reduction plans associated with both the Program and organization. SA reporting will confirm that the identified risks are managed in accordance with the provisions of the SLS-PLAN-180, Space Launch System (SLS) Program Risk and Opportunity Management Plan, and/or the software organization risk management process, and that associated action items are reported, managed, and followed through to closure. Software Assurance will identify and report risks through the SMA organization.

15.0 SOFTWARE SAFETY

In keeping with the efforts to integrate the approach to hardware and software hazard identification and resolution Software Safety (SwS) is addressed within the safety section of the SLS SMA Plan, SLS-PLAN-013.

16.0 SOFTWARE RELIABILITY

16.1 Introduction

The software reliability program provides measures for predicting, tracking and trending software reliability factors for the software development throughout the software life cycle. The QD34 organization is responsible for the software reliability. This software reliability program provides a means for continued assessment and improvement of the software design to ensure safe and reliable operation.

16.1.1 Purpose and Goal

The software reliability program will determine the probability of reliable software operation upon demand for the duration of the mission. A goal of 0.99995 is a target value for the software reliability program. Software reliability reflects the accuracy of functional requirements, design execution, and testing to identify and eliminate defects, which may propagate into critical faults affecting overall vehicle reliability and safety.

The software reliability program will model and analyze the reliability of the software to:

- Ensure requirements compliance
- Identify potential errors, defects, faults, and failures
- Identify consequence of failure
- Determine the probability of software failures causing a Loss of Mission (LOM)
- Identify and mitigate risks that affect the success of the mission and safety of the crew
- Detect design weaknesses
- Identify cost and schedule impacts

Analyses will be updated and data trended throughout each phase of the software life cycle, as described in Section 16.2 and depicted in Table 16-1.

16.1.2 Software Reliability Methodology

Software reliability analysis methods will be used to verify conformance with requirements. This will require performing functional analysis of CSCIs as well as measuring and classifying defects and duration of test time at the module level of the CSCI. Software modules described in

the CSCI software requirement specifications (SRS) includes sufficient detail to identify the modules intended purpose(s).

Software reliability analyses and modeling will be performed across the software development cycle with analysis results reported at each milestone review. Updating of the reliability data and models will continue throughout the software life cycle.

Software reliability methodology utilizes numerous methods such as Failure Modes and Effects Analysis (FMEA), software prediction and software growth models to assess the reliability and risk of the software CSCI being evaluated. The methodology includes analyzing and evaluating software and firmware for missing or inadequate requirements, poor design, improper coding methods, incorrect redundancy, insufficient maintenance and corrective action prior to final acceptance for launch use. The focus of all analyses is directed at detecting software defects which could ultimately cause or propagate into a failure that would cause a Loss of Mission (LOM) or Loss of Crew (LOC), redundancy and reliability.

16.1.2.1 Failure Modes and Effects Analysis (FMEA)

Failure Modes and Effects Analysis (FMEAs) are used to identify failure modes, causes, and effects of those failure modes that would adversely affect the operation of the vehicle. A criticality code relative to the mission phase is assigned for each failure mode. FMEAs also provide input to risk assessment and risk management activities. FMEAs will be a functional analysis and as design matures and detail is added it will be expanded to include the increased level of detail. As design changes are made the analysis will be updated to include analysis of the new configuration. The FMEA report will contain a description of the CSCIs evaluated as well as any ground rules and assumptions made in the performance of the FMEA.

16.1.2.2 Software Reliability Prediction Models

Software Reliability Prediction Models for each CSCI will be based on the SoftRel© Full-scale survey. The SoftRel© Full-scale survey is used to determine the initial reliability prediction based on factors such as Capability Maturity Model Integration (CMMI©) level, programmer experience, formal unit testing, length of test program, and other factors that are based on industry best practices. The software development organization will be surveyed to determine the current reliability prediction given the number of lines of source code, programming language, and Full-scale survey results.

16.1.2.3 Software Reliability Growth Models

Software Reliability Growth Models estimate growth across all phases of testing to acceptance. The estimation model is used to determine inherent defects, failure rates and manpower needed to complete testing to meet the target reliability value and schedule. Exponential, Logarithmic, or Non-Homogenous Poisson Point (NHPP) modeling may be utilized once testing has begun and failure distributions have been determined. The modeling type will be determined by the nature, and type of defects as well as failure rate trends. This activity begins in the software testing phase.

16.2 Software Life Cycle Reliability Tasks

This section identifies tasks by software lifecycle phase.

Table 16-1. Software Reliability Activities by Life Cycle

Activity	Phase: Software Requirements Phase	Software Preliminary Design Phase	Software Detailed Design Phase	Software Implementation	Software Test Phase	Software Operations and Maintenance Phase
Perform FMEA Analysis		●	●	●	●	●
Measure SW Reliability Predictions	●	●	●	●		
Perform Software Studies (as required)	●	●	●	●	●	●

Software being a network system, which interfaces with numerous components in various ways, will be analyzed for proper redundancy and functionality for identified mode of operation at each design phase such as unit testing, Software Development Facility (SDF), System Integration and Test Facility (SITF), System Integration Lab (SIL), and integrated vehicle testing. The analysis will be performed for each operational mode.

16.2.1 Software Reliability in the Requirements Phase

16.2.1.1 Tasks

Predict software reliability based upon industry best practices relative to CMMI© maturity level, programming language and Thousand Source Lines of Code (KSLOC) for each CSCI.

16.2.1.2 Inputs

Includes, but is not limited to the Software Requirements Specification.

16.2.1.3 Activities

The following activities are conducted in support of the System Requirements Review (SRR).

- Develop software reliability prediction analysis model to determine initial defect density, and time to reach software reliability target values.
- Review software specifications.
- Perform software reliability trade studies as required.

16.2.1.4 Products

Generation of the internally used software reliability prediction analysis is used as an input for assessing software risk as part of the integrated hazard analysis conducted by QD35. If performed, software reliability trade studies will be documented per the guidelines and format established by the SLS Program and provided to all applicable stakeholders.

16.2.2 Software Reliability in the Preliminary Design Phase

16.2.2.1 Tasks

- Ensure software functional and redundancy requirements have been met in the current design.
- Ensure software reliability predictions accurately reflect current design parameters.
- Perform software reliability trade studies as required.

16.2.2.2 Inputs

Includes, but is not limited to the Software Requirements Specification.

16.2.2.3 Activities

The following activities are conducted in support of the Preliminary Design Review (PDR).

- Develop FMEA utilizing CSCI functional descriptions as identified in the SRS. The FMEA will provide analysis to identify potential issues with software development during the requirements phase of software and firmware prior to the Preliminary Design Phase.
- Update software reliability prediction analysis model to reflect design changes.
- Perform trade studies as required

16.2.2.4 Products

Software FMEA is developed and any issues identified are worked with flight software engineering through various technical interchange meetings until resolved and the FMEA is updated. This updated FMEA is distributed to Core Stage/Boeing/SMA for integration into the flight computer FMEA. If performed, software reliability trade studies will be documented per the guidelines and format established by the SLS Program and provided to all applicable stakeholders.

16.2.3 Software Reliability in the Detailed Design Phase

16.2.3.1 Tasks

- Continue to ensure software functional and redundancy requirements have been met in the current design.

- Ensure software reliability predictions accurately reflect current design parameters.
- Perform software reliability trade studies as required.

16.2.3.2 Inputs

Includes, but is not limited to:

- Flight Computer Software Requirements Specification
- FMEAs (current versions)

16.2.3.3 Activities

The following activities are conducted in support of the Critical Design Review (CDR).

- Update FMEA to current software design maturity.
- Update software reliability prediction analysis model to reflect current software design.
- Perform software reliability trade studies as required.

16.2.3.4 Products

Software FMEA is updated and any issues identified are worked with flight software engineering through various technical interchange meetings until resolved and the FMEA is updated. This updated FMEA is distributed to Core Stage/Boeing/SMA for integration into the flight computer FMEA. The software reliability prediction analysis is used as input for assessing software risk as part of the integrated hazard analysis conducted by QD35. If performed, software reliability trade studies will be documented per the guidelines and format established by the SLS Program and provided to all applicable stakeholders.

16.2.4 Software Reliability in the Software Implementation Phase

16.2.4.1 Tasks

- Continue to ensure software functional and redundancy requirements have been met in the current design.
- Ensure software reliability predictions accurately reflect current design parameters.
- Perform software reliability trade studies as required.

16.2.4.2 Inputs

Includes, but is not limited to:

- Flight Computer Software Requirements Specification
- FMEAs (current versions)

16.2.4.3 Activities

The following activities are conducted in support of the Critical Design Review (CDR).

- Update FMEA to current software design maturity.
- Update software reliability prediction analysis model to reflect current software design.
- Perform software reliability trade studies as required.

16.2.4.4 Products

Software FMEA is updated and any issues identified are worked with flight software engineering through various technical interchange meetings until resolved and the FMEA is updated. This updated FMEA is distributed to Core Stage/Boeing/SMA for integration into the flight computer FMEA. The software reliability prediction analysis is updated and provided as input for assessing software risk as part of the integrated hazard analysis conducted by QD35. If performed, software reliability trade studies will be documented per the guidelines and format established by the SLS Program and provided to all applicable stakeholders.

16.2.5 Software Reliability in the Software Test Phase

16.2.5.1 Tasks

- Estimate reliability growth of software based on test metrics.
- Continue to ensure software functional and redundancy requirements have been met in the current design.
- Ensure software reliability predictions accurately reflect current design parameters.
- Perform software reliability trade studies as required.

16.2.5.2 Inputs

Includes, but is not limited to:

- Flight Computer Software Requirements Specification
- FMEAs (current versions)
- Software Problem Reports discovered during testing.

16.2.5.3 Activities

The following activities are conducted in support of the Software Test Readiness Review (SwTRR).

- Develop software reliability growth analysis model.
- Update FMEA to current software design maturity.

- Compare software reliability predication analysis model and software reliability growth analysis model.
- Perform software reliability trade studies as required.

16.2.5.4 Products

Generation of the internally used software growth analysis based on problem reports discovered during testing is compared against the software reliability predication analysis. This comparison is used to provide any updates to the risk assessment of the integrated hazard analysis performed by QD35. Software FMEA is updated and any issues identified are worked with flight software engineering through various technical interchange meetings until resolved and the FMEA is updated. This updated FMEA is distributed to Core Stage/Boeing/SMA for integration into the flight computer FMEA. If performed, software reliability trade studies will be documented per the guidelines and format established by the SLS Program and provided to all applicable stakeholders.

16.2.6 Software Reliability in the Operations and Maintenance Phase

16.2.6.1 Tasks

- Update software reliability growth based on problem reports discovered during testing.
- Continue to ensure software functional and redundancy requirements have been met in the current design.
- Ensure software reliability estimates accurately reflect current design.
- Perform software reliability trade studies as required.

16.2.6.2 Inputs

Includes, but is not limited to:

- Flight Computer Software Requirements Specification
- FMEAs (current versions)
- Software Test problem report data

16.2.6.3 Activities

The following activities are conducted, as required, in support of the operation and maintenance of the flight software.

- Update software reliability growth analysis model.
- Update FMEA to current software design maturity.
- Validate Software Reliability.

- Perform software reliability trade studies as required.

16.2.6.4 Products

The software growth analysis is updated, based on problem reports discovered during operations/testing, is compared against the software reliability predication analysis. This comparison is used to provide any updates to the risk assessment of the integrated hazard analysis performed by QD35. Software FMEA is updated and any issues identified are worked with flight software engineering through various technical interchange meetings until resolved and the FMEA is updated. This updated FMEA is distributed to Core Stage/Boeing/SMA for integration into the flight computer FMEA. If performed, software reliability trade studies will be documented per the guidelines and format established by the SLS Program and provided to all applicable stakeholders.

17.0 INDEPENDENT VERIFICATION AND VALIDATION

The SLS flight software development effort has been classified per NPR 7150.2 as a "CLASS A" development effort with safety-critical software and was submitted to the Office of the Chief Engineer as part of the annual software inventory. The NASA Chief Safety and Mission Assurance Officer which chairs the IV&V Board of Directors (IBD) prioritized the software inventoried and allocation of IV&V services and has selected the SLS Program for IV&V support.

The planning for the IV&V approach is defined in the Space Launch System Program Independent Verification and Validation Project Execution Plan (IPEP).

APPENDIX A
ACRONYMS AND ABBREVIATIONS AND GLOSSARY OF TERMS

A1.0 ACRONYMS AND ABBREVIATIONS

CDR	Critical Design Review
CM	Configuration Management
CMMI	Capability Maturity Model Integration
CSCI	Computer Software Configuration Item
BIT	Built in Test
DRD	Data Requirements Description
FC	Flight Computer
FCA	Functional Configuration Audit
FMEA	Failure Modes and Effects Analysis
FSW	Flight Software
IV&V	Independent Verification and Validation
LOC	Loss of Crew
LOM	Loss of Mission
OJT	On the Job Training
MSFC	Marshall Space Flight Center
NHPP	Non-Homogenous Poisson Point
PCA	Physical Configuration Audit
PDR	Preliminary Design Review
SA	Software Assurance
SAP	Software Assurance Plan
SCM	Software Configuration Management
SCMP	Software Configuration Management Plan
SDF	Software Development Facility
SDL	Software Development Library
SDP	Software Development Plan
SIL	System Integration Laboratory
SITF	System Integration and Test Facility
SMA	Safety and Mission Assurance
SRB	System Review Board
TBD	To Be determined

A2.0 GLOSSARY OF TERMS

No Unique terms need defining.

APPENDIX B
COMPLIANCE MATRIX

NASA-STD-8739.8	SAP	Comments
6.0	N/A	N/A - Paragraph Header
6.1	N/A	N/A - Paragraph Header
6.1.1	Entire document	The SAP references MSFC S&MA/SA applicable OIs.
6.1.2	Task 3.3.15	Addresses non-development software
6.1.3	Section 3 and subs; 15.0; 16.0; 17.0	In general the SAP addresses Software Quality and Software V&V, Software Safety and Software Reliability with references made to the IV&V Plan
6.1.4	17.0	
6.1.5	5.2	
6.2	N/A	N/A - Paragraph Header
6.2.1	3.0	
6.2.2	3.3; 3.4	
6.2.3	3.0; 3.1	
6.2.4	3.0.6	
6.2.5	3.3 and Subs; 6.1	
6.2.6	8.0 and subs; 14.0	
6.3	N/A	N/A - Paragraph Header
6.3.1	Entire document	Table 1-1 identifies the Life Cycle Activities
6.3.2	Entire document	
6.3.2.1	1.0	Reference SAP DRD
6.3.2.2	1.0	Reference SAP DRD
6.3.2.3	1.0	Reference SAP DRD
6.4	N/A	N/A - Paragraph Header
6.4.1	12.4	Addresses changes to SAP
6.4.2	12.4	Addresses changes to SAP
6.5	3.3.6	
6.6	N/A	N/A - Paragraph Header
6.6.1	12.0 and subs	
6.6.2	8.0 and subs	Process Audit Reports include suggested corrective action
6.7	N/A	N/A - Paragraph Header

NASA-STD-8739.8	SAP	Comments
6.7.1	6.2; 12.1.5	
6.7.1.a	6.2; 12.1.5	
6.7.1.b	6.2; 12.1.5	
6.7.1.c	6.2; 12.1.5	
6.7.1.d	6.2; 12.1.5	
6.7.1.e	6.2; 12.1.5	
6.7.1.f	6.2; 12.1.5	
6.7.1.g	6.2; 12.1.5	
6.8	N/A	N/A - Paragraph Header
6.8.1	13.0	
6.8.2	13.0	
6.8.3	13.0	
6.8.4	13.0	
6.8.5	13.0	
6.9	N/A	N/A - Paragraph Header
6.9.1	N/A	No known subcontractors associated with this development
6.9.2	N/A	No known subcontractors associated with this development
7.0	N/A	N/A - Paragraph Header
7.1	N/A	N/A - Paragraph Header
7.1.1	N/A	N/A – Lead in Sentence
7.1.1.1	3.3.8	
7.1.1.2	3.3.8; 3.3.10; 3.3.12; 3.3.14;	
7.1.1.3	3.3.8	
7.1.1.4	3.3.8; 6.1	
7.1.1.5	3.3.16; 3.3.17; 7.0	
7.1.1.6	3.3.16; 3.3.24	
7.1.1.7	5.1	
7.1.1.8	3.3.8	
7.1.1.9	3.3.8; 6.1; 3.3.16; 3.3.17; 70	
7.1.1.10	6.1; 6.2	

NASA-STD-8739.8	SAP	Comments
7.1.1.11	8.0	
7.1.2	N/A	N/A – Lead in Sentence
7.1.2.1	1.1	
7.1.2.2	8.0	
7.1.2.3	4.0	
7.1.2.4	6.1	
7.1.2.5	3.3.10 – 3.3.16; 3.3.21; 3.3.23; 3.3.24; 3.3.27 – 3.3.30	
7.1.2.6	5.1	
7.2	N/A	N/A - Paragraph Header
7.2.1	15.0	
7.2.2	15.0	
7.2.3	15.0	
7.2.4	15.0	
7.3	N/A	N/A - Paragraph Header
7.3.1	16.0	
7.3.2	16.0	
7.3.3	16.0	
7.3.4	16.0	
7.3.5	16.0	
7.4	N/A	N/A - Paragraph Header
7.4.1	6.1	
7.4.2	6.1	
7.4.3	7.0	
7.4.4	5.1	
7.4.5	7.0	
7.4.6	3.3.18; 3.3.19; 3.3.21	
7.5	N/A	N/A - Paragraph Header
7.5.1	17.0	

NASA-STD-8739.8	SAP	Comments
7.5.2	17.0	
7.5.3	17.0	
7.5.4	17.0	
7.5.5	17.0	

APPENDIX C
OPEN WORK

C1.0 TO BE DETERMINED

Table C1-1 lists the specific To Be Determined (TBD) items in the document that are not yet known. The TBD is inserted as a placeholder wherever the required data is needed and is formatted in bold type within carets. The TBD item is sequentially numbered as applicable (i.e., **<TBD-001>** is the first undetermined item assigned in the document). As each TBD is resolved, the updated text is inserted in each place that the TBD appears in the document and the item is removed from this table. As new TBD items are assigned, they will be added to this list in accordance with the above described numbering scheme. Original TBDs will not be renumbered.

Table C1-1. To Be Determined Items

TBD	Section	Description

C2.0 TO BE RESOLVED

Table C2-1 lists the specific To Be Resolved (TBR) issues in the document that are not yet known. The TBR is inserted as a placeholder wherever the required data is needed and is formatted in bold type within carets. The TBR issue is sequentially numbered as applicable (i.e., **<TBR-001>** is the first unresolved issue assigned in the document). As each TBR is resolved, the updated text is inserted in each place that the TBR appears in the document and the issue is removed from this table. As new TBR issues are assigned, they will be added to this list in accordance with the above described numbering scheme. Original TBRs will not be renumbered.

Table C2-1. To Be Resolved Issues

TBR	Section	Description